UC Santa Cruz
UC Santa Cruz Previously Published Works

Title

Fault Detection for Covered Conductors With High-Frequency Voltage Signals: From Local Patterns to Global Features

Authors

Chen, Kunjin
Vantuch, Tomáš
Zhang, Yu
et al.

Fault Detection for Covered Conductors With High-Frequency Voltage Signals: From Local Patterns to Global Features

Kunjin Chen, Tomáš Vantuch, Yu Zhang, *Member, IEEE*, Jun Hu, *Member, IEEE*, and Jinliang He, *Fellow, IEEE*

Abstract—The detection and characterization of partial discharge (PD) are crucial for the insulation diagnosis of overhead lines with covered conductors. With the release of a large dataset containing thousands of naturally obtained high-frequency voltage signals, data-driven analysis of fault-related PD patterns on an unprecedented scale becomes viable. The high diversity of PD patterns and background noise interferences motivates us to design an innovative pulse shape characterization method based on clustering techniques, which can dynamically identify a set of representative PD-related pulses. Capitalizing on those pulses as referential patterns, we construct insightful features and develop a novel machine learning model with a superior detection performance for early-stage covered conductor faults. The presented model outperforms the winning model in a Kaggle competition and provides the state-of-the-art solution to detect real-time disturbances in the field.

Index Terms—Covered conductor, partial discharges, clustering methods, gradient boosting trees.

I. INTRODUCTION

Covered conductors have been widely used for medium voltage overhead lines in forested or dissected terrain areas because of their higher operational reliability and reduced land use [1]. Compared with uninsulated overhead lines, the interphase touch of conductors or the contact with tree branches of covered conductors does not lead to an immediate short-circuit fault [2]. Nevertheless, a persistent contact with a tree branch may degrade conductor insulation over time and develop into a fault that hampers the normal operation of a power distribution system. Therefore, it is favorable to detect such faults to prevent full deterioration of the insulation system. A major indicator is the partial discharge (PD) activity induced by the fault [1], [3], [4].

Generally speaking, the PD pattern can be represented as an impulse component of the current or voltage signal generated by PD activity rooted from the insulation deterioration [5], [6]. The correct evaluation of PD patterns on the medium voltage overhead lines with covered conductors can be used for early detection of developing faults, thus improving the system's safety and reliability [3], [7]–[9]. Depending on the PD pattern evaluation from the current and voltage signals, two principles are currently used to detect covered conductor faults. The first principle is to evaluate the PD pattern as an impulse component of the current measured by Rogowski coil [7]–[9]. The second principle is to evaluate the PD pattern as an impulse component of the voltage signal measured by a capacitive divider [3].

The data used in this work comes from a metering sensor that is based on a single layer coil wrapped around the covered conductor to acquire the voltage signal of electric stray field along the covered conductor [10]. High cost efficiency is an advantage of this solution compared to its competitors [8], which leads to its broader deployment. However, to make it really work, an effective pattern recognition method is required to recognize faulty-related PD activities from its acquired signals. Several studies were brought to analyse the signal characters using fuzzy theory [11], mathematical chaos [12], complex networks [2] or machine learning models [4]. To bring more attention on this problem, in 2018, a dataset which contained a large number of signal measurements was published on Kaggle, the world's largest data science collaboration platform [13]. The intent of the organizers was to attract worldwide researchers and data scientists to examine the signal data and to develop effective pattern recognition models for PD detection.

A successful fault detection model for covered conductors has to correctly handle the following factors:
- The measurements are not synchronised with sinusoidal waveform shapes. Hence, accurate sine shape removal must be included.
- Deployment in nature-like environment introduces a significant and yet varying external background noise

interference that interpolates false pulse patterns. A proper signal thresholding procedure is needed [14].

- A large number of various PD pattern types [12] increases the problem's complexity, which requires a robust detection model.
- Naturally, a significantly higher number of fault-free signal samples results in a highly imbalanced dataset [15].

In this article, we develop a novel fault detection model for covered conductors based on data-driven PD pattern exploitation. In the literature, signal processing and machine learning techniques have been widely used in the PD pattern detection in general [16], [17] and were combined with various feature extraction techniques like wavelet decomposition [18], recurrence quantification analysis [19], fractal features [20] or image processing. Two common types of PD pattern representations are phase-resolved PD patterns and individual PD pulse shapes. For phase-resolved PD patterns, the extracted features generally describe the distribution of quantities such as phase angles, charges, and counts [21], [22]. In addition, features can also be extracted by analyzing parameters related to pulse shapes [23], [24]. Extracted features are generally used as inputs for the machine learning models including random forest, support vector machines (SVM), gradient boosting machines [1], [4].

The fault detection task is not about classifying the source of individual PD pulses, nor about classifying PD sources by using overall PD patterns accumulated over multiple cycles. The nature of early fault detection along with the complexity of signals collected by low-cost sensors from real environments [1] requires that a good approach should be able to identify fault-related pulse patterns and aggregate information at the signal sample level. However, existing feature extraction methods mainly focus on individual pulses and aggregate the pulse-level features via simple statistics, which may be insufficient to reveal the fault-related information.

Our approach differentiates from existing methods mainly by its feature construction procedure. The major contribution of the present work is that we approach the pulse patterns in a data-driven manner and use clustering to bridge the gap between the locality of pulses and the desirability of building global features. The clustering results are not used for identifying the sources of pulses, but for constructing discriminative features. Different from features constructed using all pulses, cluster-specific features are not affected by pulses within other clusters. Novel features including template-matching degree and intra-cluster concentration degree are proposed. The results based on the same dataset and evaluation criteria from the Kaggle competition also make the model comparable and reproducible.

The organization of the paper is as follows. In Section II, detailed steps of the proposed approach are provided, which include signal pre-processing, pulse identification and feature construction. Section III covers the detailed numerical results and discussions. Finally, conclusions are drawn in Section IV.

II. THE PROPOSED FAULT DETECTION APPROACH

In this section, we first describe the dataset and the overall diagram of the proposed approach. Steps including signal
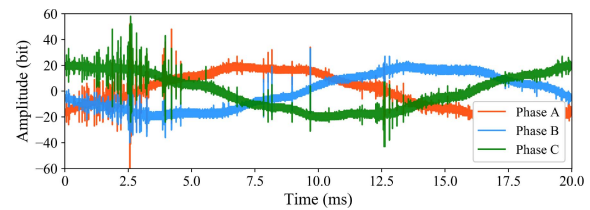

Fig. 1. An example of a three-phase signal labelled as faulty. The label is decided according to examination of a fault that was recorded after the timestamp of this signal sample, together with visual inspection of the pulses within the range from 0 to 5 ms.

pre-processing, pulse identification, and pulse clustering are implemented before the construction of features. The classifier used for fault detection and the evaluation metric are also introduced.

A. Problem Description and Overview of the Approach

In Fig. 1, an example of a three-phase high-sampling-rate voltage signal measured from a medium voltage overhead line is presented. The time span of the signals is 0.02 seconds and the sampling rate is 40 MHz (i.e., each signal has 800000 time steps). The fault detection task aims to determine whether developing covered conductor faults appear in the signals. To construct the training dataset, the one-cycle signals are labelled by a combination of expert observation and field examination [10]. For the faulty class, only the signals corresponding to the time range prior to the full development of faults are included. Therefore, a model that is able to classify faulty and non-faulty signals correctly can help prevent the covered conductors from being burned completely, which may lead to serious consequences like wildfires. A major challenge for the task is that some of the non-faulty signals appear to have faulty patterns [10]. It is worth noting that unlike lab experiments, no prior knowledge of fault-related PD patterns or background noise interference is available from the dataset. Therefore, a data-driven approach developed for this setting has a high potential to be generalizable to other measurement devices, interference sources, etc. More details of the fault detection task can be found in [1], [3], [10].

A total of 2904 three-phase signals are collected for the training dataset, while 2916 three-phase signals (the same as the *private leaderboard* in the Kaggle competition) are used for testing. Out of the 8712 single-phase signals in the training dataset, 8187 signals (93.97%) are non-faulty, which indicates that the dataset is quite imbalanced. Although faulty and non-faulty labels are manually assigned to each single-phase signal, we consider each three-phase signal as a whole and treat it as faulty if at least one of the single-phase signals is faulty. More details of the dataset can be found in [10].

The diagram of the proposed approach is illustrated in Fig. 2, where the implementation steps are as follows:

- The three-phase signals first go through a phase correction process so that the signal of each phase is aligned to the same phase angle.
- The signals are flattened so that the pulses' amplitudes in the signals can be compared. A threshold is estimated for each signal to filter out the background noise.
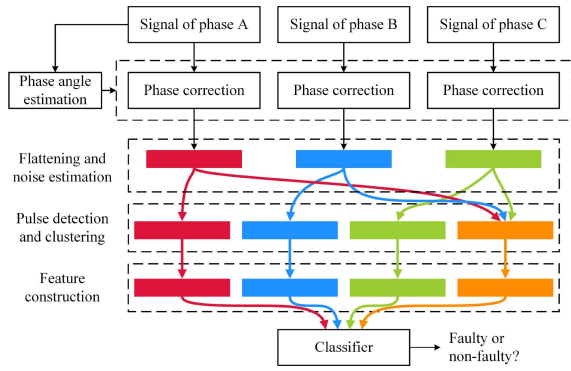
Fig. 2.   The diagram of the proposed fault detection approach.

Fig. 3.   An example of the signal flattening process.

- The pulses are detected by finding the points with the highest amplitudes within their neighbourhoods and the waveforms surrounding the pulses are clustered to reveal the waveform patterns. Clustering is performed for the three phases individually as well as combined.
- Features are constructed based on the detected pulses and the clustering results. The features are then used to train a classifier that is able to distinguish between faulty and non-faulty signals.

B. Signal Pre-Processing

The information related to PD is generally contained within the pulses in the signals. Thus, an important step before the implementation of machine learning models is to detect and analyze the pulses. Specifically, we propose a data-driven approach to reveal the characteristics of the pulses.

Two steps of pre-processing are needed before we can detect pulses from the signals. The first step is phase angle correction and signal flattening, and the second step is noise level estimation. For phase angle correction, we use discrete Fourier transform (DFT) to extract the power frequency component in the signals and align the signals to have a phase angle of zero degree at the first time step. We then use the Savitzky-Golay filter [25] to fit the low frequency part of a signal, which is then subtracted from the signal so that the signal is flattened. Specifically, a value y_j^* in the filtered signal can be calculated as

$$y_j^* = \sum_{i=(1-m)/2}^{(m-1)/2} C_i y_{j+i}, \tag{1}$$

where m is the window size and C_i is the ith coefficient. The details for computing the coefficients can be found in [26]. In Fig. 3, the process of filtering the signal with the Savitzky-Golay filter and signal flattening is illustrated.

The method for noise level estimation is described in Algorithm 1. The inputs of the algorithm are the flattened signal s, the number of signal sections N_{noise}, the length of each signal section L_{noise}, the threshold for the number of sections to cover N_{cover}, the maximum noise level C_{max}, and the incremental step size for noise level C_{step}. For simplicity, we estimate one global noise level for each single-phase signal. An illustration of the noise estimation algorithm is illustrated
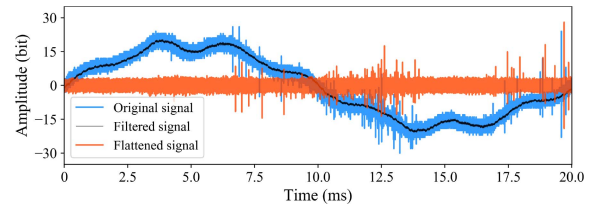

Algorithm 1: Noise Level Estimation

Input: $s, N_{\text{noise}}, L_{\text{noise}}, N_{\text{cover}}, C_{\text{max}}, C_{\text{step}}$
Output: a

1   Sample N_{noise} sections ($\bar{s}_1$ to $\bar{s}_{N_{\text{noise}}}$) with length L_{noise} equidistantly from s
2   **for** $i = 1$ **to** N_{noise} **do**
3     $m_i \leftarrow \max(\text{abs}(\bar{s}_i))$
4   **end**
5   **for** $i = 1$ **to** $C_{\text{max}}/C_{\text{step}}$ **do**
6     $k \leftarrow 0$
7   **for** $j = 1$ **to** N_{noise} **do**
8   **if** $m_j < i \cdot C_{\text{step}}$ and $m_j >= (i-1) \cdot C_{\text{step}}$ **then**
9         $k \leftarrow k + 1$
10   **if** $k > N_{\text{cover}}$ **then**
11          $a \leftarrow i \cdot (C_{\text{step}} + 1)$
12   **break**
13   **end**
14   **end**
15   **end**
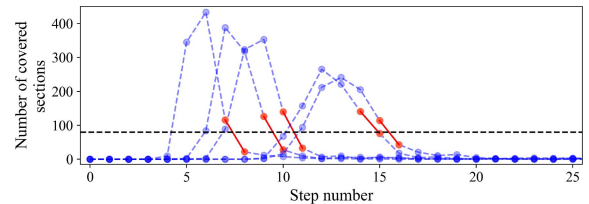16   **end**
17   **return** a

Fig. 4.   An illustration of the noise estimation method with a threshold of 80. Five examples are randomly selected and the line sections crossing the threshold from top to bottom are highlighted.

in Fig. 4. It can be seen that the threshold $N_{\text{cover}} = 80$ helps find a step number that has a low m_i for each signal, such that the amplitudes covered in the section are not likely to belong to background noise. In Fig. 5, we present examples of flattened signals together with estimated noise levels. It is observed that the estimated noise levels are able to cover the low-amplitude background noise without covering the high-amplitude pulses in the signals.

C. Pulse Identification

Specifically, a point in a signal is considered a pulse if it has the highest absolute amplitude in its neighbourhood. A fast and robust way to locate the pulses in the signals is needed
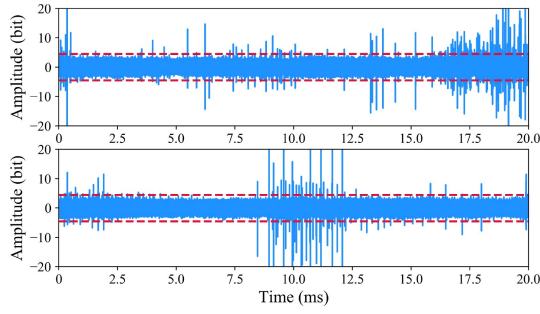
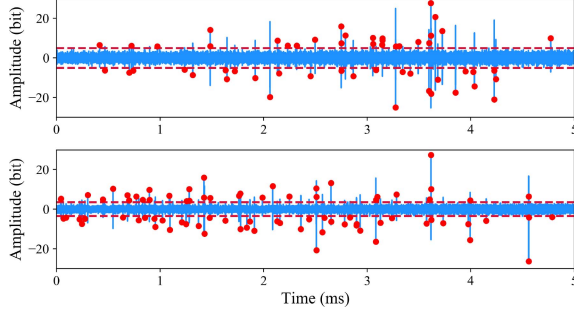Fig. 5. Examples of estimated noise level on flattened signals.

Fig. 6. Examples of detected pulses on flattened signals.

Algorithm 2: Pulse Identification and Processing

Input: $s, N_{sort}, N_{top}, N_{mask}, N_{local}, a, C_{mag}$
Output: P

1   $s_{mask} \leftarrow \text{abs}(s), s_{abs} \leftarrow \text{abs}(s), P' \leftarrow \emptyset, P \leftarrow \emptyset$
2   **for** $i = 1$ **to** 3 **do**
3   Divide s_{mask} into N_{sort} sections ($\bar{s}_1$ to $\bar{s}_{N_{sort}}$)
4   **for** $j = 1$ **to** N_{sort} **do**
5       $P'_j \leftarrow$ indexes of largest N_{top} values in $\bar{s}_j$
6   Add P'_j to P'
7   **for** $p' \in P'_j$ **do**
8         $s_{mask}[p' - N_{mask} : p' + N_{mask}] \leftarrow 0$
9   **end**
10   **end**
11   **end**
12   **for** $p \in P'$ **do**
13   **if** $s_{abs}[p] = \max(s_{abs}[p - N_{local} : p + N_{local}])$ **then**
14   **while** Cond_1 is satisfied **do**
15         $p \leftarrow p - 1$
16   **end**
17   **if** Cond_2 is satisfied **then**
18         $p \leftarrow p - 2$
19   **else if** Cond_3 is satisfied **then**
20         $p \leftarrow p - 3$
21   **end**
22   **if** $s_{abs}[p] > a$ and $s_{abs}[p] < 50$ **then**
23   Add p to P
24   **end**
25   **end**
26   **end**
27   return P

---

to facilitate the subsequent feature extraction and classification steps. Concretely, a two-stage pulse detection method is adopted, where the first stage finds a set of pulse candidates, and the second stage verifies if the candidates are valid pulses. The pulse detection method is described in detail in Algorithm 2. Besides s, the algorithm inputs also include the number of sections to equally split the signal N_{sort}, the number of largest amplitude values to keep N_{top}, the neighbourhood radius for masking N_{mask}, the neighbourhood radius of local maximum N_{local}, the estimated noise level a, and the threshold for magnitude comparison C_{mag}.

The first stage of the algorithm ensures that the pulses are not densely distributed within a small time section of the signals and that pulses of both high and low amplitudes can be selected. In the second stage, all pulse candidates collected from the first stage are verified to guarantees that the absolute amplitudes of the pulses are the maxima within their neighbourhoods. Examples of detected pulses can be seen in Fig. 6. In addition to the verification of amplitude, three conditions for relocation of pulses are described as $\text{Cond}_i:s[p - i]s[p] < 0$ and $s_{abs}[p - i] > C_{mag}s_{abs}[p], i = 1, 2, 3$. Relocation of the pulses is crucial for the construction of features based on the shapes of the waveforms surrounding the pulses. The conditions above are able to find high-amplitude points a few time steps before the pulse candidates.

D. Pulse Clustering and Global Features Construction

The previously-mentioned pulse detection method is able to locate the pulses with high absolute amplitudes within their neighbourhoods, but as we are more concerned about the waveforms surrounding the pulses, it is then of great importance to investigate the shapes of the waveforms. A straightforward way to compare the shapes of different

waveforms is to align the waveforms with the pulses as the anchor point. That is, we normalize the waveforms surrounding the pulses by dividing the amplitude of the corresponding pulses and align the waveforms such that the pulses are located at the same position. Further, we expect that typical pulse waveform patterns can be revealed by clustering of the waveforms. The patterns will later serve to produce a set of finely tuned features describing the signal for the purpose of its classification.

The clustering of pulses is carried out for individual phases as well as all phases combined. Up to N_{sample} pulses are randomly sampled from the signal of each phase for clustering so that the clustering process is not excessively influenced by signals with a large number of pulses. Specifically, k-means clustering with k-means++ initialization [27] is applied to the waveforms surrounding the identified pulses. Given a waveform x, the length of the waveform is $N_{before} + N_{after} + 1$, where N_{before} and N_{after} are the number of time steps before and after the anchor point. After the clustering of pulse waveforms is performed, the centroids of the clusters can be calculated for further analysis.

The pulses detected in the signals and the clustering results cannot be directly used as inputs to machine learning models. Thus, it is necessary to build compact global features that can be used to distinguish between faulty and non-faulty
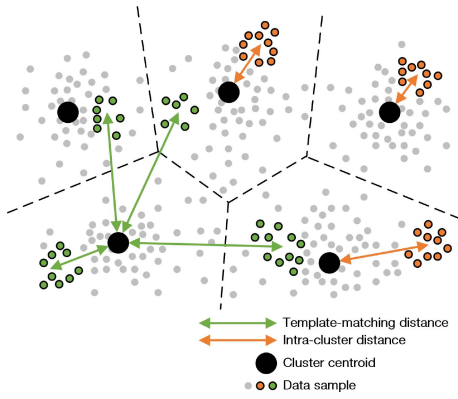
Fig. 7. An illustration of template-matching features (green dots) and intra-cluster concentration features (orange dots).

signals. With the clustering results for individual phases and all phases combined, building features based on pulses in a specific phase (or all phases) or in a specific cluster becomes possible. Some traditional and trivial features include count of pulses, average height of pulses, and standard deviation (SD) of height of pulses. These features, however, are insufficient in reflecting the shapes of the pulses. Thus, two additional deep feature groups, namely, template-matching degree, and intra-cluster concentration degree, are introduced. Specifically, the features are constructed by calculating the rooted mean squared error (RMSE) between pulse waveforms and templates (for template-matching degrees) or cluster centroids (for intra-cluster concentration degrees).

Given the ith template $\boldsymbol{p}_i$, the corresponding template-matching feature r_i is calculated as

$$r_i = \frac{1}{N_{\text{pulse}}} \sum_{j=1}^{N_{\text{pulse}}} \text{RMSE}(\boldsymbol{x}_j, \boldsymbol{p}_i), \qquad (2)$$

where N_{pulse} is the number of pulses detected in a three-phase signal. The templates are not necessarily cluster centroids but can be built based on the centroids. While the template-matching features are calculated using all pulses, the intra-cluster concentration degrees are obtained using pulses assigned to each cluster centroid. For each cluster centroid $\boldsymbol{\mu}_i$, the intra-cluster concentration feature d_i is obtained by

$$d_i = \frac{\sum\limits_{\boldsymbol{x}_j:\text{NC}(\boldsymbol{x}_j)=i} \text{RMSE}(\boldsymbol{x}_j, \boldsymbol{\mu}_i)}{\sum\limits_{\boldsymbol{x}_j:\text{NC}(\boldsymbol{x}_j)=i} 1}, \qquad (3)$$

where NC is the centroid assignment function given the cluster centroids. For phase-specific cluster centroids, the pulses are also limited in the phases. The features of all-phase cluster centroids use pulses from all phases.

We use Fig. 7 to demonstrate the concepts of template-matching and intra-cluster concentration features. Suppose the green data samples belong to a specific signal, then the template-matching features are calculated based on all the samples (indicated by the green arrows). Further, suppose the orange data samples belong to another signal, we can calculate the intra-cluster concentration features using the samples within each voronoi cell (indicated by the orange arrows).

In the proposed approach, we construct two types of features, namely, *all-pulse* features and *cluster-specific* features. Specifically, all-pulse features are constructed using all pulses identified across three phases, while cluster-specific features are constructed with pulses within pulse clusters (both for individual phases and all phases combined). As each pulse is assigned to one of the clusters, features constructed for a specific cluster are not affected by pulses in other clusters. In addition, template-matching features are built for all-pulse features only and intra-cluster concentration features are exclusive for cluster-specific features. Pulse count, average height, and SD of height features are used for both types of features. When all the features are constructed, we put them together and build classifiers based on the features.

E. Classifier and Evaluation Metric

In this article, we use light gradient boosting machine (lightGBM) [28] as the classifier. The lightGBM model is an efficient implementation of gradient boosting decision tree (GBDT) [29], a gradient boosting framework built upon tree-based learning algorithms. Specifically, the GBDT model is an addictive ensemble of decision trees that are trained in a sequential manner, and each tree is trained based on the negative gradients of the loss function. The implementation of lightGBM is based on XGBoost [30], which optimizes the loss function using second-order approximation. The lightGBM model is suitable for the features we have constructed because of the following reasons:

1) Tree-based models are in general insensitive to the values of the features, thus only the raw features are needed. Specifically, the values of a feature are ordered and put into bins by building a histogram. For each split, values within bins on two sides of the split point are separated into two new leaves.
2) The lightGBM model automatically decides which feature to use when splitting a leaf node. Thus, the importance of a feature can be reflected by the number it is used or the gain of the feature.

Specifically, binary-crossentropy is used as the loss function for lightGBM for the binary classification task in this work. As non-faulty samples account for over 93% of the training dataset, the training of the classification model may be hampered by this imbalance. One effective method to tackle class imbalance is over-sampling the minority class (faulty) near the borderline between two classes, which can lead to an improved decision boundary with a mild increase in dataset size. In this work, borderline synthetic minority oversampling technique (SMOTE) with SVM is used to create synthetic faulty samples. Specifically, SMOTE-SVM [31] first trains an SVM on the original dataset to approximate the borderline area with support vectors. New faulty samples are synthesized near support vectors for the faulty class so that the desired ratio of the number of faulty samples to the number of non-faulty samples, α, is achieved (interested readers are referred to [31] for the details). The SMOTE-SVM is used as a final ingredient for our proposed approach.

The Matthews correlation coefficient (MCC) is used to evaluate the fault classification performance of different

approaches, and it is defined as

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \quad (4)$$

where *TP* is the true positive rate (the rate samples predicted as positive are indeed true samples, similarly hereinafter); *TN* the true negative rate; *FP* the false positive rate; and *FN* the false negative rate. The MCC is considered a good evaluation metric for binary classification tasks on imbalanced datasets [32].

III. RESULTS AND DISCUSSION

In this section, we describe the implementation details of the proposed approach and present the results for fault detection. The performance of the proposed approach is enhanced by analyzing feature importance values. Additionally, a discussion on the number of clusters is provided.

A. Implementation Details for Signal Pre-Processing and Pulse Identification

The implementation details of the methods for signal pre-processing and pulse identification are elaborated as follows:
1) Signal flattening: the Savitzky-Golay filter uses a window size of 99 and a polynomial order of 3.
2) Noise level estimation: N_{noise} and L_{noise} are set to 1000 for the sampling of signal sections. The threshold N_{cover} is set to 80. Considering the possible range of noise level, C_{max} is set to 15 and the step size C_{step} is -0.5.
3) Pulse identification and processing: We set N_{sort} and N_{top} to 20 and 100, respectively. The values for N_{mask} and N_{local} are 50 and 25. The magnitude coefficient C_{mag} is set to 0.5.

B. Analysis of Pulse Patterns and Clustering of Pulses

Clustering of waveforms surrounding the pulses plays an important role in the proposed approach. In this subsection, we first use clustering of pulse waveforms from phase B to explain why the conditions in Algorithm 2 are added. Specifically, the inspection of the clustering results indicate that relocation of the pulses is needed to reveal the characteristics of the waveforms surrounding pulses. After the pulses are identified, we perform clustering to the pulse waveforms from individual phases as well as from all three phases, producing four groups of clustering results that are further exploited to construct features. Specifically, the clustering results are obtained with $N_{sample} = 100$.

We use the waveforms taken from phase B signals that are faulty to illustrate the pulse waveform inspection procedure. As a first step, the 30-time-step waveforms ($N_{before} = 15$ and $N_{after} = 14$) are clustered into three groups using the k-means algorithm with $k = 3$. Waveforms of two clusters are shown in Fig. 8 (a) with different colors, and it is observed that both clusters have two sub-modes within them. We shift the anchor points of the waveforms in phase B two or three steps ahead if either of the two time steps has an amplitude lower than -0.5. It is clearly observed in Fig. 8 (b) that the modification of the anchor points helps reveal the two types
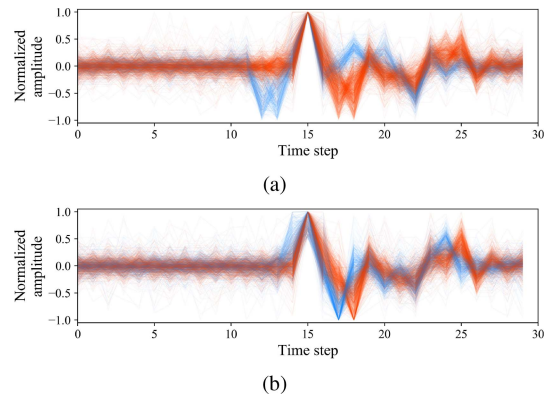

Fig. 8. Two clusters of the waveforms of pulses for phase B before and after modifying the anchor points: (a) before and (b) after.
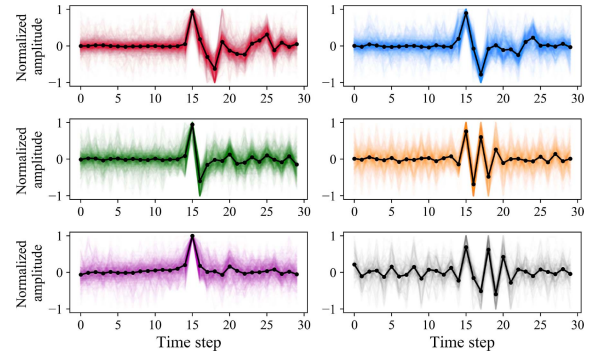

Fig. 9. Centroids for the waveforms of different clusters for phase B when $k = 6$. Random waveform samples of the clusters are also illustrated.

of waveforms that have an opposite pulse two or three steps after the anchor point. In addition, a similar modification of anchor points for waveforms that has an amplitude lower than -0.5 one time step before the anchor point is implemented. As multiple pulses with opposite signs may exist, the modification process can repeat for more than once.

The previous analysis is only for waveforms from faulty signals, which does not cover the complete characteristics of the different waveform types in phase B. A larger value of k can be selected to discover more types of waveforms. Here, a simple heuristics is used to determine the value of *k*. As The two waveform types in Fig. 8 (b) are quite similar to each other, waveforms of the two types are prone to be found within the same cluster if waveforms from non-faulty signals are included and the value of k is small. Thus, we increase the value of *k* and set it to the first value that is able to separate waveforms from the two types. Specifically, $k = 6$ satisfies the criterion and we use the same value for the clustering of waveforms for phase A and C.

The six waveform types for phase B signals are illustrated in Fig. 9. The centroids of the clusters are plotted with black lines. Apart from the two types discussed earlier, the four remaining types are also quite distinctive. A visualization of the clustering result for phase B is shown in Fig. 10. Principal component analysis (PCA) [33] is used to reduce the dimensionality of the waveforms and the first three principal components are chosen. For ease of presentation, 15% of the pulse waveforms are randomly sampled for the visualization. In addition, we use a few examples to show how the different
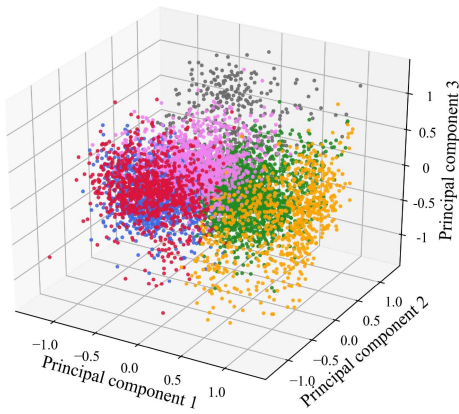
Fig. 10. Visualization of the clustering result for phase B using three principal components.
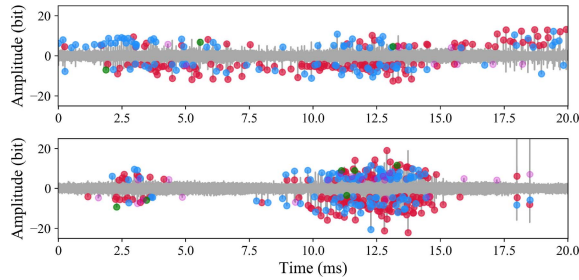

Fig. 11. Examples of flattened signals for phase B that are faulty. The detected pulses are marked with colors corresponding to the types in Fig. 9.
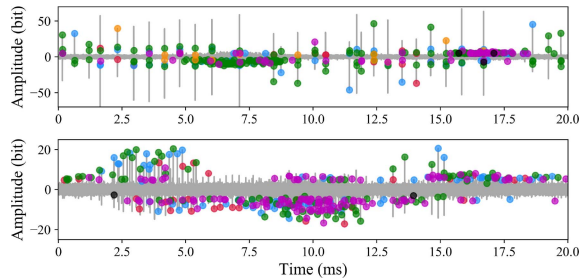

Fig. 12. Examples of flattened signals for phase B that are not faulty. The detected pulses are marked with colors corresponding to the types in Fig. 9.

waveform types can be used to distinguish between faulty and non-faulty signals. In Fig. 11 and Fig. 12, we use dots with different colors to indicate the location of different types of waveforms in the signals. The colors for the dots are identical for the waveforms of different types in Fig. 9. It can be observed from the figures that faulty and non-faulty signals differ in terms of the types of waveforms surrounding pulses and the density of pulses.

The clustering of pulse waveforms for phase A and C can be implemented in a similar way, and the centroids of the clusters for the two phases are shown in Fig. 13. Although the orders of the centroids for the two phases are not matched, it can be observed that there is a one-to-one relationship between the centroids. It is worth pointing out that the pulse waveforms in each cluster appear in both faulty and non-faulty signals. To better illustrate this, the average ratios of pulse counts in the clusters to the number of pulses in faulty and non-faulty signals for each phase are given in Fig. 14. It can be seen that although some clusters account for higher proportions in faulty
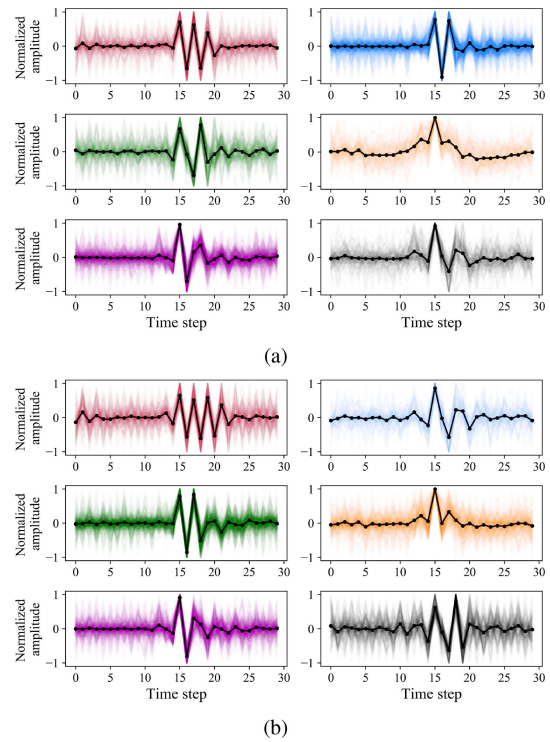
Fig. 13. Centroids for the waveforms of different clusters for (a) phase A and (b) phase C when $k = 6$.
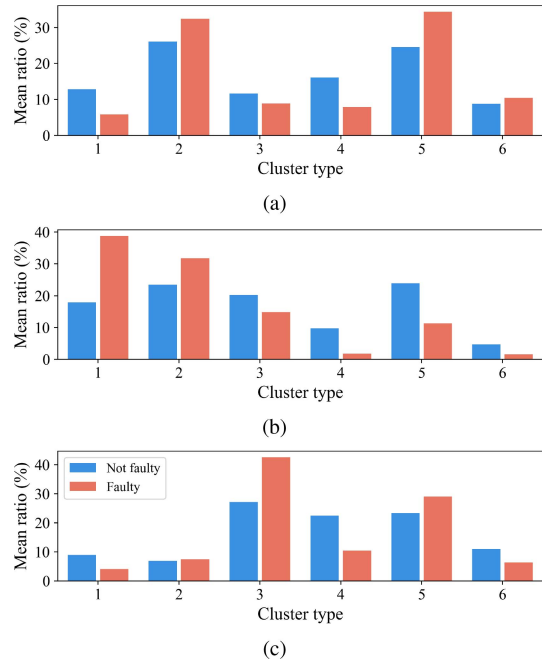
Fig. 14. The average ratio of pulse counts in each cluster to the number of pulses in faulty and non-faulty signals for (a) phase A, (b) phase B, and (c) phase C.

signals, these clusters still share relatively large proportions in non-faulty signals. Therefore, it is more important to use the clustering results for the construction of distinctive features instead of using them as fault indicators directly. In addition, it is impractical to classify the source of all the pulses appearing in the field signals. The data-driven approach based on

Fig. 15. Centroids for the waveforms of the detected pulses of different clusters for all phases when $k = 15$. The ordering of the types is from left to right, then from top to bottom.

Fig. 16. The values of normalized SSE for (a) individual phases and (b) all phases.

Fig. 17. The templates used for constructing the template-matching features.

clustering can provide clues about early-stage faulty conditions without explicitly identifying the fault-related pulses.
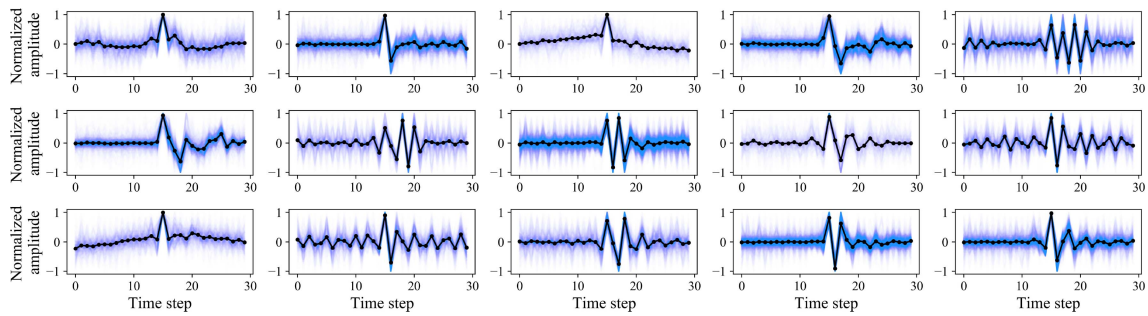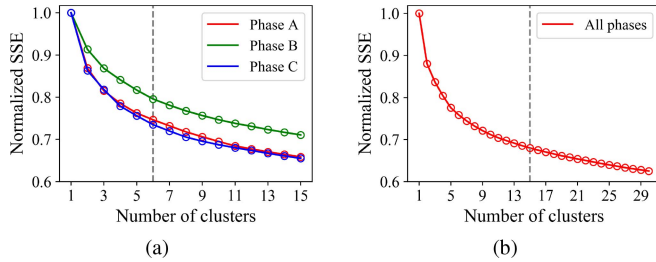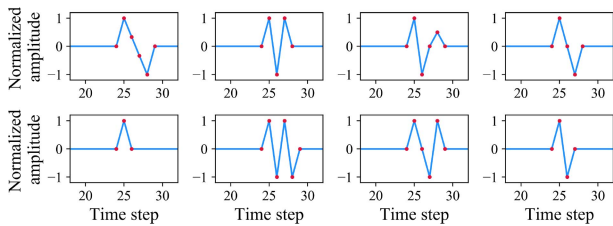
In addition to phase-specific clustering of waveforms, the clustering of waveforms from all phases is also conducted. In Fig. 15, the waveforms of the clusters with $k = 15$ are presented. For most of the centroids in the figure, it is not difficult to find similar centroids for individual phases.

The sum of squared error (SSE) (i.e., the sum of squared distance from each pulse to its centroid) can be used to choose proper values of k for k-means clustering. In Fig. 16, we plot the values of SSE for the several cases with different values of k. We normalize the SSE values so that the first value is always 1. A simple method for choosing k is picking a k so that the slope of the curve after the kth point becomes relatively small and stable. While it is hard to determine optimal values of k from the curves, we can see that $k = 6$ for individual phases and $k = 15$ for all phases (dashed lines) roughly satisfy the criterion of the heuristic method.

C. Description of Features and Classifier

As is introduced in II-D, we construct all-pulse and cluster-specific features separately. The descriptions for the all-pulse

features are provided in Table I. The features in Table I are referenced from the Kaggle's 1st place solution, and the templates for template-matching features are shown in Fig. 17. The length of the templates is 50 time steps. Initially, only the first template in Fig. 17 is introduced in the Kaggle's 1st place solution. We add 7 more features to this feature group based on the centroids illustrated in Fig. 15. In Table II, we describe the cluster-specific features in detail. Each feature group has 33 features, 15 of which are for all phases combined and 18 of which are for individual phases (6 for each phase).

Similar to the implementation in the Kaggle's 1st place solution, a total of 125 lightGBM models are trained with 25 different random seeds and 5-fold cross-validation [34]. Generally speaking, 5-fold cross-validation means that we split the training dataset into 5 disjoint parts and leave one of the parts out at a time as the validation set. In order to compare with the Kaggle's 1st place solution, we follow its cross-validation scheme that uses 60% of the data for training, 20% for validation, and the remaining 20% for testing. After determining the features and the hyper-parameters, five models are trained in the 5-fold manner. With a different random seed, the split of the training dataset changes so that more models can be trained to facilitate ensemble learning [35]. The final logits are calculated by averaging the logits of all models and the threshold for the logits is estimated by training samples.

The experiments of the proposed approach are conducted in Python 3.6 using CPUs. With a single thread, the average time lengths for feature extraction (three phases) and inference of 125 LightGBM models are 0.0156 and 0.0055 seconds, respectively. Using multiple threads for calculation further reduces the computation time. For instance, it takes less than 0.0008 seconds for the inference of the LightGBM models when 8 threads are used. Hence, our efficient approach is suitable for real-time fault detection applications.

D. Fault Detection Results and Comparison

We compare the performance of the proposed model with other approaches. Spccifically, the 1st place solution[1] is also based on features constructed on pulses, but the majority of the features are pulse counts and statistics of pulse heights.

[1]https://www.kaggle.com/mark4h/vsb-1st-place-solution

TABLE I
DESCRIPTION OF ALL-PULSE FEATURES

Feature name	Description of the feature	Number of features
Pulse count	Number of pulses for quadrants one and three, quadrants two and four, and all quadrants	3
Average height	Average pulse height for quadrants one and three	1
SD of height	SD of pulse heights for quadrants one and three	1
Template-matching degree	Average RMSE between the waveforms and templates for quadrants one and three	8

TABLE II
DESCRIPTION OF CLUSTER-SPECIFIC FEATURES BASED ON PULSES OF DIFFERENT CLUSTERS

Feature name	Description of the feature	Number of features
Pulse count	Number of pulses for each cluster	33
Average height	Average pulse height for each cluster	33
SD of height	SD of pulse heights for each cluster	33
Intra-cluster concentration degree	Average RMSE between the cluster centroids and waveforms in the clusters	33

TABLE III
COMPARISON OF THE MCC VALUES FOR DIFFERENT APPROACHES

Model	MCC	Precision	Recall
Reference [4]	0.589	0.573	0.643
Kaggle 5th place	0.697	-	-
Kaggle 2nd place	0.715	-	-
Kaggle 1st place	0.719	0.616	**0.867**
Proposed	**0.735**	**0.669**	0.841

The 2nd place solution[2] is a combination of CatBoost models [36] and recurrent neural network models. The 5th place solution[3] combines lightGBM models with recurrent neural networks and convolutional neural networks. The approach in [4], which uses a series of traditional features including count, width, and height of pulses, is also implemented as a benchmark. In Table III, the metrics of the different models are presented. It is seen in the table that the proposed approach outperforms the benchmark as well as the top solutions for the competition on MCC. The metrics of precision $\frac{TP}{TP+FP}$ and recall $\frac{TP}{TP+FN}$ are also shown in Table III. It is observed in the table that the proposed approach has much higher precision. Specifically, there are 377 faulty signals in the test dataset, 327 out of the 531 signals predicted as faulty are actually faulty for the Kaggle's 1st place solution, and 317 out of the 474 signals predicted as faulty are true positives for the proposed approach. In a nutshell, the proposed approach is less likely to produce false positive predictions while the recall rates of the two approaches are roughly at the same level.

E. Performance Enhancement With Feature Analysis

The LightGBM model can evaluate the importance of features by calculating the total gain of each feature. Since the final classification result is given by the combination of multiple LightGBM models, the feature importance of each individual model are added to obtain the final feature

---

[2]https://www.kaggle.com/c/vsb-power-line-fault-detection/discussion/86616#latest-501584

[3]https://www.kaggle.com/c/vsb-power-line-fault-detection/discussion/85170#latest-500367

---

importance. Fig. 18 shows the logarithmic values of feature importance, which are given by the model with an MCC of 0.735. In order to facilitate the comparison of different types of features, they are divided into four categories, and their positions in the feature importance ranking are marked with different colors. The four types of features are count features, RMSE features, SD features and height features. Specifically, RMSE features include both template-matching features and intra-cluster concentration features.

It can be seen in Fig. 19 that the three most important features are pulse count features, followed by three RMSE features (two of which are template-matching features). The importance values of SD features and average height features are lower overall. The average height features generally have the lowest importance values. It is expected that the distributions of faulty and non-faulty samples are quite different for the most important features. Fig. 19 shows the distributions of faulty and non-faulty samples for the six most important features. The three pulse count features are all cluster-specific, of which the first and third features use pulses from all phases, and the second feature is a single-phase feature (A-5 indicates the fifth cluster of phase A). The fourth and fifth features are template-matching features, and the sixth feature is an intra-cluster concentration feature that uses pulses from three phases. It is observed that the distributions of the top pulse count features of faulty samples are significantly higher than those of non-faulty samples, while the distributions of top RMSE features are significantly lower than those of non-faulty samples. Thus, these features can be used to distinguish the faulty and non-faulty samples.

A difference between template-matching features and intra-cluster concentration features is that the template-matching features only use pulses within quadrant one and quadrant three of the signals, while the intra-cluster concentration features use all pulses. An obvious reason is that the numbers of pulses used for intra-cluster concentration features are much smaller than template-matching features, which may increase the chance of overfitting if the number of pulses is further reduced. A detailed analysis on the reason that template-matching features use pulses within quadrant one and
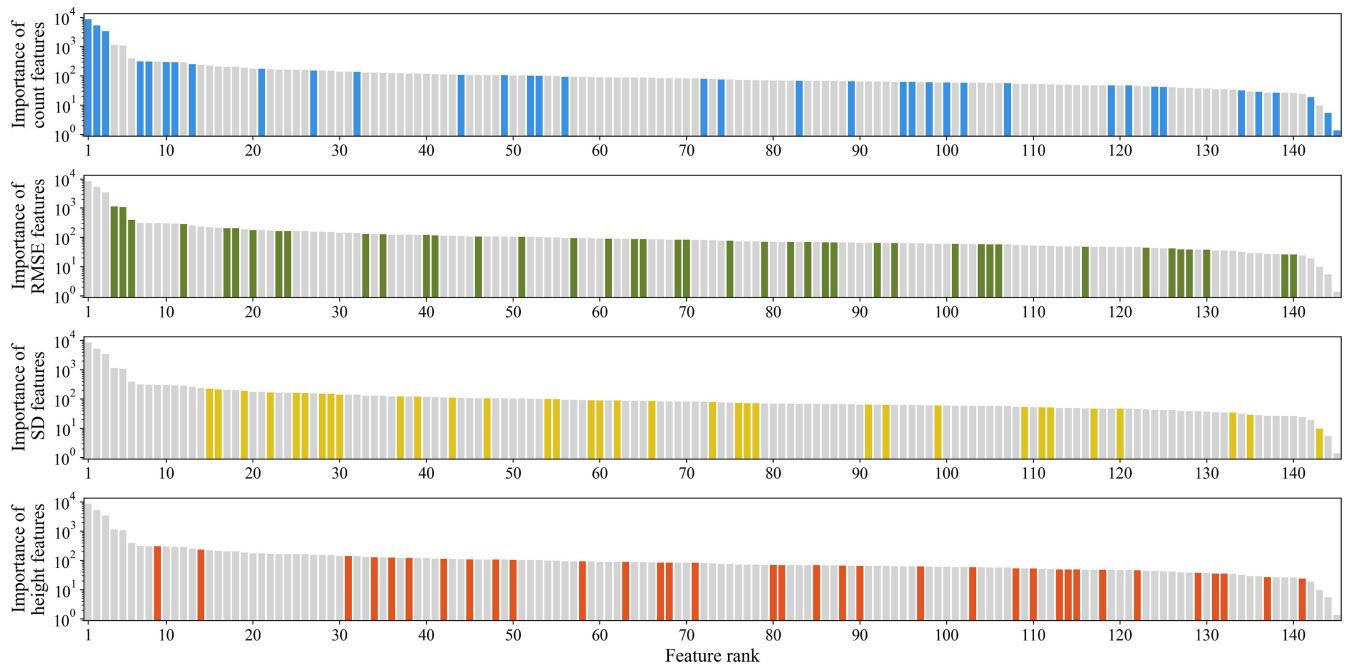
Fig. 18.   The feature importance values of different feature groups provided by the LightGBM model.
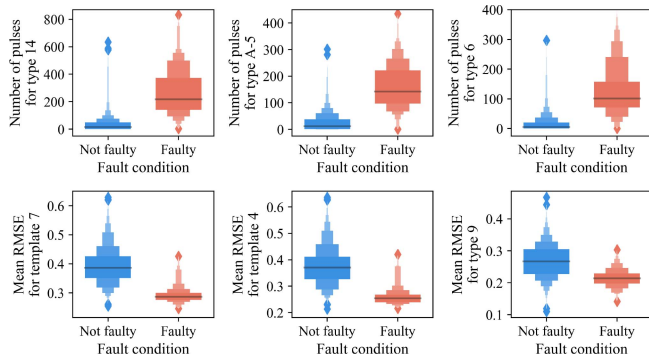

Fig. 19.   Distributions of six features with the highest feature importance values. The top three features are pulse count features and the other three are RMSE features.

Fig. 20.   Boxplots for phase-resolved distributions of the number of pulses for types 13, 4, 15, and 6 in Fig. 15 (top to bottom).

Fig. 21.   Distributions of three SD features with the highest feature importance values.

quadrant three is given in Fig. 20, which shows the distributions of pulse counts within phase-resolved segments for the clusters corresponding to the four most important template-matching features. Each signal is evenly divided into 20 signal segments where the pulses are counted separately. Specifically, the four clusters are clusters 13, 4, 15, and 6, which correspond to templates 7, 4, 3, and 1, respectively. For the first two features, the pulse counts in quadrant one and quadrant three (marked with red line) are generally more than quadrant two and quadrant four for faulty samples, while non-faulty samples generally have more pulses in quadrant two and quadrant four. This distributional characteristic indicates that these two features can be more discriminative if the counts are taken for quadrant one and quadrant three only.

Fig. 21 shows the distributions of three SD features with the highest feature importance, ranking 15, 16, and 19, respectively. Compared with the top-ranked features, the discriminative power of these features for faulty and non-faulty

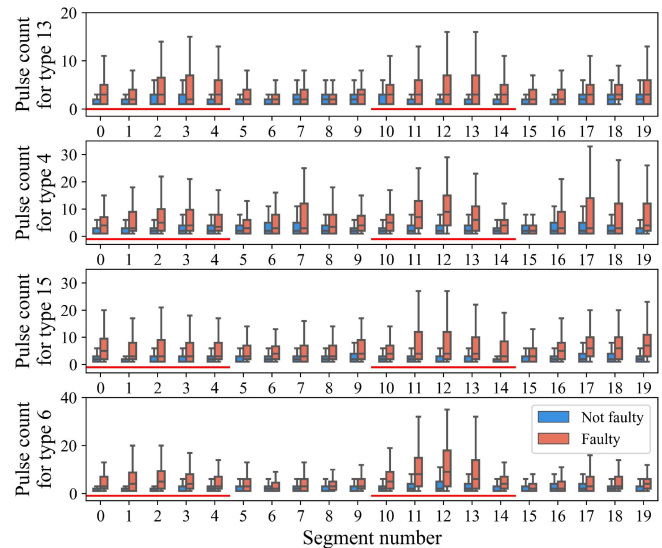samples has decreased. Nevertheless, the distributions of faulty samples and non-faulty samples for these features are still visibly different. This indicates that these features can still provide useful information during the learning process of DTs.
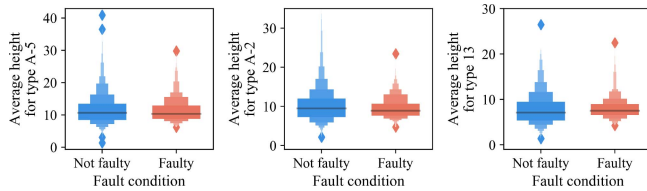
Fig. 22. Distributions of three average height features with the highest feature importance values.

TABLE IV
COMPARISON OF RESULTS FOR THE PROPOSED MODEL

Detail	MCC	Precision	Recall
I	0.735	0.669	0.841
I w/o average height features	**0.740**	**0.671**	**0.844**
II	**0.750**	0.693	0.838
II (all-pulse features)	0.722	0.615	**0.881**
II (cluster-specific features)	0.740	**0.717**	0.788
III ($\alpha = 0.1$)	0.755	0.695	0.846
III ($\alpha = 0.15$)	**0.766**	**0.706**	**0.854**
III ($\alpha = 0.2$)	0.751	0.705	0.825

In Fig. 22, the distributions of three average height features with the highest feature importance are plotted (the features ranked 9, 14, and 31, respectively). Although the importance ranks of the first two features are relatively high, it can be seen that the distributions of faulty and non-faulty samples are quite similar (almost identical mean values). Compared with other types of features, these features are more likely to lead to overfitting of the models. To justify this argument, we can remove the average height features and evaluate the performance of the model.

As mentioned above, the average height features have lower feature importance values in general. Even the features with the highest importance values have lower distinguishability between faulty and non-faulty samples than other feature groups. Therefore, in a new experiment setting, average height features are removed and the LightGBM models are retrained. Next, we change the cross-validation scheme to the standard leave-one-out version and estimate the threshold using validation data, which improves the model's performance (also without average height features). Finally, over-sampling of faulty samples with SMOTE-SVM is implemented in addition to the improved cross-validation scheme.

A comparison of the results for the proposed approach is presented in Table IV, where model I represents the model with all features, model II represents the model with the improved cross-validation scheme, and model III represents model II with SMOTE-SVM applied to each split of cross-validation. For model II, additional results are added for models with only all-pulse features or cluster-specific features. As shown in the table, the average height features have a negative impact on the performance of the classification model when other groups of features are used. Shifting to the standard leave-one-out cross-validation also improves the model's performance. Over-sampling the faulty samples with SMOTE-SVM with an α between 0.1 and 0.2 generally lifts the MCC, and model III with $\alpha = 0.15$ has the highest MCC.

Nevertheless, as a single threshold is used for the evaluation of the metrics, the trade-off between precision and recall leads to the discussion in the following subsection.

F. Performance Comparison With Varying Thresholds

For each test sample, the classification model first gives the sample a "fault probability" between 0 and 1, and then determines whether the sample is faulty or not based on a threshold obtained by cross-validation. In fact, as the threshold moves from 0 to 1, the values of both TP and FP gradually decrease, while the value of FN gradually increases. Therefore, the recall value decreases monotonically with the increase of the threshold. The precision value, whose shape is determined by the distribution of the output probabilities of the samples, tends to increase with the increase of the threshold. In order to have a clear understanding of the performance of different models, we draw curves of various metrics obtained by varying the threshold within a wide range.

Fig. 23 shows a comparison of the proposed model and the 1st place solution on Kaggle for several metrics when the threshold varies from 0.2 to 0.8. As the figure suggests, the proposed model II and Kaggle's 1st place solution have similar recall values, but model II has a significant improvement on the precision value. In fact, the threshold estimated for model II is 0.434, while the threshold for the 1st place solution on Kaggle is 0.350. The result shows, however, that model II performs better on MCC for all thresholds ranging from 0.2 to 0.8. The proposed model III significantly improves the recall metric with an estimated threshold of 0.566. As expected, over-sampling the faulty samples is prone to producing more false positive samples and fewer false negative samples. In summary, model III has the highest MCC at its estimated threshold and outperforms other models when the threshold is greater than 0.5.

G. Discussion on the Number of Clusters

In Section III-B, it is mentioned that the number of clusters can be determined by two methods, namely, the heuristic method and the method based on SSE evaluation. Although pulse waveform clustering is only an intermediate process of feature construction, the analysis in Section III-B implies that a proper choice of the number of clusters can help reveal the pulse patterns.

Specifically, increasing the number of clusters results in a more finely-divided space of pulse waveforms, so that the constructed features can capture the distribution of pulse waveform data more precisely. However, a large number of clusters reduces the number of pulse waveforms within each cluster, thus it is more likely to cause the problem of overfitting. Therefore, it is necessary to examine the effect of the number of clusters on the model's performance. We conduct experiments with different number of clusters and compare the MCC values of each case. Model II with only cluster-specific features is used to have a rigorous comparison.

The MCC values of the proposed model with different cluster numbers are shown in Fig. 24. The labels for different cases represent the numbers of clusters for pulses from individual
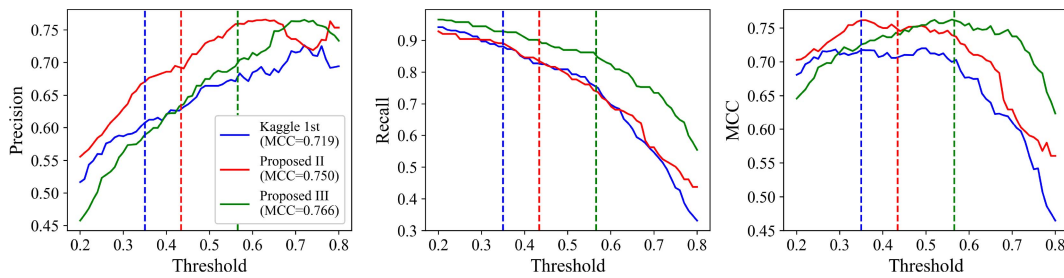
Fig. 23.   Comparison between the proposed model and the 1st place solution on Kaggle with varying thresholds.
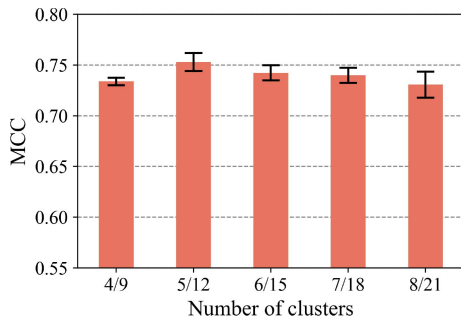

Fig. 24.   The MCC values of the proposed model using cluster-specific features with different cluster numbers. Each value is averaged from five trials and the error bars indicate the range of one SD above and below the average values.

phases and all phases, respectively. Five trials are conducted for each case (clustering results differs from trial to trial as the pulses used for clustering are randomly sampled) and the SD values are calculated to yield the error bars. With only cluster-specific features available, the proposed model is able to achieve MCC values larger than 0.73 for all cases. Two conclusions can be drawn from Fig. 24.

- The number of clusters has a certain influence on the performance of the model. However, Fig. 24 shows that the performance does not vary much for different numbers of cluster within a certain range. Nevertheless, the methods of determining the number of clusters provided in Section III-B are able to find the numbers that result in relatively high MCC values.
- Increasing the number of clusters will increase the variance of the MCC values. This is probably due to the fact that the number of pulses assigned to each cluster decreases, so the model is more prone to overfitting.

IV. CONCLUSION

In this article, we propose an efficient approach for faulty and non-faulty covered conductor classification based on PD patterns. With a meticulous design of data pre-processing procedure, pulses related to PD patterns are revealed. In-depth analysis of pulse patterns and clustering of pulse waveforms enable the construction of cluster-specific and all-pulse features. In addition to traditional features, two new features, namely, template-matching degree, and intra-cluster concentration degree are proposed. An in-depth analysis of feature importance reveals that average height features can be excluded for the fault detection task. Over-sampling the

faulty data samples near the borderline is proved to be useful in dealing with the imbalance of faulty and non-faulty samples. Extensive results show that the proposed approach, which achieves an MCC value of 0.766, has a significant edge over existing methods. Results with varying classification thresholds further corroborate the superiority of the proposed approach. Furthermore, our generic design can be applied to many other pattern recognition tasks.

Our future goal will be focused on performance enhancement by using deep neural networks. Deeper evaluation of representative pulse waveforms can be utilized to identify whether they come from noise interference, PD effect or other sources. Such a procedure may further improve the explainability of the proposed approach.

REFERENCES

[1] M. Krátký, S. Mišák, P. Gajdoš, P. Lukáš, R. Bača, and P. Chovanec, "A novel method for detection of covered conductor faults in medium voltage overhead line systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 543–552, Jan. 2018.

[2] T. Vantuch, J. Gaura, S. Mišák, and I. Zelinka, "A complex network based classification of covered conductors faults detection," in *Proc. Euro-China Conf. Intell. Data Anal. Appl.*, 2016, pp. 278–286.

[3] S. Mišák and V. Pokorný, "Testing of a covered conductor's fault detectors," *IEEE Trans. Power Del.*, vol. 30, no. 3, pp. 1096–1103, Jun. 2015.

[4] S. Mišák, J. Fulnecek, T. Vantuch, T. Buriánek, and T. Jezowicz, "A complex classification approach of partial discharges from covered conductors in real environment," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 24, no. 2, pp. 1097–1104, Apr. 2017.

[5] P. Wang, G. C. Montanari, and A. Cavallini, "Partial discharge phenomenology and induced aging behavior in rotating machines controlled by power electronics," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 7105–7112, Dec. 2014.

[6] G. J. Paoletti and A. Golubev, "Partial discharge theory and technologies related to medium-voltage electrical equipment," *IEEE Trans. Ind Appl.*, vol. 37, no. 1, pp. 90–103, Jan./Feb. 2001.

[7] G. M. Hashmi and M. Lehtonen, "Effects of rogowski coil and covered-conductor parameters on the performance of PD measurements in overhead distribution networks," in *Proc. 16th Power Syst. Comput. Conf.*, 2009, pp. 14–18.

[8] G. M. Hashmi, M. Lehtonen, and M. Nordman, "Modeling and experimental verification of on-line PD detection in MV covered-conductor overhead networks," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 17, no. 1, pp. 167–180, Feb. 2010.

[9] G. Hashmi, M. Lehtonen, and M. Nordman, "Calibration of on-line partial discharge measuring system using Rogowski coil in covered-conductor overhead distribution networks," *IET Sci. Meas. Technol.*, vol. 5, no. 1, pp. 5–13, Jan. 2011.

[10] S. Mišák, J. Fulneček, T. Vantuch, and L. Prokop, "Towards the character and challenges of partial discharge pattern data measured on medium voltage overhead lines," in *Proc. IEEE 20th Int. Sci. Conf. Elect. Power Eng.*, Kouty nad Desnou, Czechia, 2019, pp. 1–4.

[11] M. Prilepok and T. Vantuch, "Partial discharge pattern classification based on fuzzy signatures," in *Proc. Int. Conf. Intell. Inf. Technol. Ind.*, 2017, pp. 254–264.

[12] M. Lampart, T. Vantuch, I. Zelinka, and S. Mišák, "Dynamical properties of partial-discharge patterns," *Int. J. Parallel Emergent Distrib. Syst.*, vol. 33, no. 5, pp. 474–489, 2018.

[13] *VSB Power Line Fault Detection*. Accessed: Nov. 14, 2019. [Online]. Available: https://www.kaggle.com/c/vsb-power-line-fault-detection

[14] S. Sriram, S. Nitin, K. Prabhu, and M. Bastiaans, "Signal denoising techniques for partial discharge measurements," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 12, no. 6, pp. 1182–1191, Dec. 2005.

[15] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[16] N. Sahoo, M. Salama, and R. Bartnikas, "Trends in partial discharge pattern classification: A survey," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 12, no. 2, pp. 248–264, Apr. 2005.

[17] W. J. K. Raymond, H. A. Illias, A. H. A. Bakar, and H. Mokhlis, "Partial discharge classifications: Review of recent progress," *Measurement*, vol. 68, pp. 164–181, May 2015.

[18] X. Ma, C. Zhou, and I. Kemp, "Interpretation of wavelet analysis and its application in partial discharge detection," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 9, no. 3, pp. 446–457, Jun. 2002.

[19] S. Kanakambaran, R. Sarathi, and B. Srinivasan, "Identification and localization of partial discharge in transformer insulation adopting cross recurrence plot analysis of acoustic signals detected using fiber bragg gratings," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 24, no. 3, pp. 1773–1780, Jun. 2017.

[20] L. Satish and W. Zaengl, "Can fractal features be used for recognizing 3-D partial discharge patterns," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 2, no. 3, pp. 352–359, Jun. 1995.

[21] E. Lalitha and L. Satish, "Wavelet analysis for classification of multi-source PD patterns," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 7, no. 1, pp. 40–47, Feb. 2000.

[22] K. Firuzi, M. Vakilian, B. T. Phung, and T. R. Blackburn, "Partial discharges pattern recognition of transformer defect model by LBP & HOG features," *IEEE Trans. Power Del.*, vol. 34, no. 2, pp. 542–550, Apr. 2019.

[23] F. Alvarez, J. Ortego, F. Garnacho, and M. Sanchez-Uran, "A clustering technique for partial discharge and noise sources identification in power cables by means of waveform parameters," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 23, no. 1, pp. 469–481, Feb. 2016.

[24] A. R. Mor, L. C. Heredia, and F. Muñoz, "New clustering techniques based on current peak value, charge and energy calculations for separation of partial discharge sources," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 24, no. 1, pp. 340–348, Feb. 2017.

[25] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.

[26] R. W. Schafer, "What is a Savitzky–Golay filter? [lecture notes]," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 111–117, Jul. 2011.

[27] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 1027–1035.

[28] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3146–3154.

[29] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Math. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[30] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794.

[31] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *Int. J. Know. Eng. Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011.

[32] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, 2020.

[33] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[34] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *Proc. IEEE Int. Conf. Adv. Comput.*, Bhimavaram, India, 2016, pp. 78–83.

[35] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.*, 2000, pp. 1–15.

[36] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6638–6648.

Kunjin Chen received the B.Sc. and Ph.D. degrees in electrical engineering from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in 2015 and 2020, respectively.

His research interests include a variety of machine learning applications in power systems.

Tomáš Vantuch received the Ph.D. degree in computer science from the VŠB—Technical University of Ottava, Czech Republic, in 2018.

He has published a number of articles in peer reviewed journals and conference proceedings. He actively participates in several national and international projects. His current work includes implementation of smart grid technologies using prediction models and bio-inspired methods as well as utilization of machine learning models for industrial applications.

Yu Zhang (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Minnesota, Minneapolis, MN, USA, in 2015.

He is an Assistant Professor with the ECG Department, UNC Santa Cruz. Prior to joining USMC, he was a Postdoctoral with UNC Berkeley and Lawrence Berkeley National Laboratory. His research interests span the broad areas of cyborphysical systems, smart power grids, optimization theory, machine learning, and big data analytic.

Jun Hu (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in July 1998, July 2000, and July 2008, respectively.

He is currently an Associate Professor with the Department of Electrical Engineering, Tsinghua University. His research fields include overdosage analysis in power system, sensors and big data, dielectric materials, and surge arrester technology.

Jinliang He (Fellow, IEEE) received the B.Sc. degree in electrical engineering from Wuhan University of Hydraulic and Electrical Engineering, Wuhan, China, the M.Sc. degree in electrical engineering from Chongqing University, Chongqing, China, and the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 1988, 1991, and 1994, respectively.

He became a Lecturer in 1994, and an Associate Professor in 1996, with the Department of Electrical Engineering, Tsinghua University. From 1997 to 1998, he was a Visiting Scientist with Korea Electrotechnology Research Institute, Changwon, South Korea, involved in research on metal oxide varistor and high voltage polymeric metal oxide surge arresters. From 2014 to 2015, he was a Visiting Professor with the Department of Electrical Engineering, Stanford University, Pale Alto, CA, USA. In 2001, he was promoted as a Professor with Tsinghua University, where he is currently the Chair with High Voltage Research Institute. He has authored five books and 400 technical papers. His research interests include overclothes and EMU in power systems and electronic systems, lightning protection, grounding technology, power apparatus, and dielectric material.